



**Fakultät für
Informatik und Mathematik**

Bachelorarbeit

über das Thema

**Entwurf und Implementierung eines Chrome Plugins zur
automatischen Anreicherung von Webseiten mit kulturellen
Inhalten**

Autor: Mathias Möller
moellerm@fim.uni-passau.de

Prüfer: Prof. Granitzer

Abgabedatum: 24.08.2015

I Kurzfassung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Abstract

Das ganze auf Englisch.

II Inhaltsverzeichnis

I	Kurzfassung	I
II	Inhaltsverzeichnis	II
III	Abbildungsverzeichnis	IV
IV	Abkürzungsverzeichnis	V
1	Einleitung und Motivation	1
2	Related Work	3
2.1	Unterschiede und Gemeinsamkeiten zu JITIR-Agents	3
2.2	Vergleich mit existierenden JITIR Agents	4
2.2.1	Remembrance Agent	4
2.2.2	Margin Notes	5
2.2.3	Watson	5
2.2.4	EEXCESS	5
2.3	Text Retrieval Algorithmen	5
2.4	Unterschiede und Gemeinsamkeiten zu „Automatic help systems“ (z.B. Microsoft Office Assistant - & Domain spezifisch)	5
3	Konzept	6
3.1	Warum kein Proactiver JITIR-Agent?	6
3.2	Anzeige der Ergebnisse	6
3.3	Erklärung der Such-Anfragen Generierung	6
3.4	Anpassen der Suchanfrage durch den Nutzer	6
3.5	Verbesserung der Suchanfrage z.B. durch maschinelles Lernen	6
4	Implementierung	6
4.1	Verwendung von AngularJS für alle Komponenten des Plugins	6
4.2	Bau der GUIs	6
4.3	Einbindung der REST-Services	6
5	Evaluierung?	6
6	Future Work	6
6.1	Alternative Algorithmen zur Textanalyse	6
6.2	Implementierung einer automatischen Suchanfragen-Verbesserung durch maschinelles Lernen	6
6.3	Verbesserung der Ergebniss-Güte	7
6.4	Anpassung der Anwendung auf mobile Nutzung	7
7	Conclusion	7
8	Quellenverzeichnis	8

Anhang

I

III Abbildungsverzeichnis

IV Abkürzungsverzeichnis

1 Einleitung und Motivation

Suchmaschinen gehören heutzutage zu den meistbesuchten Seiten im Internet. Google¹ ist mit 2.000.000.000.000 Suchanfragen pro Jahr (TODO: Quelle) nicht nur die beliebteste Suchmaschine, sondern auch die meist besuchte Seite im Internet. Auch die chinesische Suchmaschine Baidu² ist in den oberen fünf Plätzen vertreten [AI].

- Auf XX% der Webseiten wird über eine Suchmaschine zugegriffen
- Menschen holen Informationen über Suchmaschinen, nicht mehr aus Bibliotheken oder Lexika
- Woher Quellen?

Um so mehr sie an Bedeutung gewinnen, umso größer ist der Bedarf an bessern Suchmaschinen [Law00].

Doch der Funktionsumfang klassischer Suchmaschinen ist begrenzt. Sie können den Kontext des Benutzers in ihren Queries? nicht miteinbeziehen. Weiterhin verlangen sie vom Nutzer, seine gegenwärtige Arbeit einzustellen, die Webseite der Suchmaschine aufzurufen und eine Suchanfrage zu formulieren. So wird die Konzentration auf den eigentlichen Arbeitsschritt gestört. Im Schnitt sind nur 67% der Suchanfragen an Google erfolgreich (TODO: Quelle). Das heißt in circa einem Drittel der Fälle bleibt der Aufruf der Suchmaschine ohne die erwarteten Erfolge. Zu der verlorenen Zeit addiert sich auch noch die Zeit, die der Nutzer braucht um sich seine Arbeit wieder ins Kurzzeitgedächtnis zu holen.

Um diesen Problemen entgegen zu wirken wurden neue Wege zur Informationsgewinnung entwickelt. Eine Möglichkeit sind so genannte Just-in-time Information Retrieval Agents (JITIR-Agents) [RM00]. Sie beobachten im Hintergrund den Kontext des Benutzers und versuchen aus den so erhaltenen Informationen eine Suchanfrage an eine Datenbank oder ein Recommender-System zu schicken. Die gewonnenen Informationen werden dann möglichst unaufdringlich dem Benutzer angezeigt. Er kann sich nun entscheiden diese Informationen genauer zu betrachten oder mit seiner Arbeit fortzufahren. Die kognitive Belastung bleibt hierbei sehr gering. JITIR-Agents reduzieren auf diese Weise enorm den Aufwand Informationen zu finden [RM00].

Durch ihre Funktionsweise sind sie jedoch nicht so exakt wie klassische Suchmaschinen, da sie nur "erraten" können, was den Benutzer gerade interessiert. Wenn ein Benutzer einer genauen Vorstellung von der Suchanfrage oder den Ergebnissen hat, haben klassische Suchmaschinen Vorteile gegenüber den JITIR-Agents [RM00].

¹<https://www.google.com/>

²<http://www.baidu.com/>

Ein weiteres Problem des Internets ist es, dass die Standards, die die simple und leicht skalierbare Architektur des Netzwerks ermöglichen, den Einsatz von fortgeschrittenen Hypermedia-Technologien verhindern [Bou99]. Eine Möglichkeit, das Internet um Funktionen zu erweitern ist Web Augmentation (WA). Diaz [Día12] beschreibt WA als den Versuch, statt eine neue Technologie zu entwickeln, neue Funktionalität auf eine gerenderte Webseite zu setzen:

In some sense, WA is to the Web what Augmented Reality is to the physical world: layering relevant content/layout/navigation over the existing Web to customize the user experience. [Día12]

Solche WA-Tools interagieren mit einem Web Server, Http-Proxy oder dem Browser der Nutzer, um so Inhalte oder Navigationselemente direkt in die angezeigte Webseite einzufügen. Auf diese Weise erlauben sie es, die limitierten Möglichkeiten des World Wide Webs zu Gunsten des Nutzers anzureichern (Quelle: Kenneth Anderson oder Bouvin).

Ein Beispiel für ein solches WA-Tool sowie für ein JITIR System ist das EEXCESS Chrome Plugin³. Es analysiert die aktuell besuchten Seiten des Nutzers und schlägt ihm auf Grund der erlangten Daten am Rand des Browser-Fensters weiterführende Quellen aus der Europeana-Datenbank⁴ vor.

Ziel dieser Bachelorarbeit ist es, auf Basis des EEXCESS Plugins ein Chrome Plugin zu entwerfen und zu implementieren, welche eine Webseite in einzelne Paragraphen aufteilt und zu jedem dieser Paragraphen kulturelle Inhalte der Europeana-Datenbank vorschlägt. Dabei soll das Design des Plugins helfen, die bei der in Kapitel 5 erläuterten Evaluierung des EEXCESS Plugins aufgetauchten Probleme zu beheben und Schwachstellen zu verbessern.

Im zweiten Teil dieser Arbeit wird auf vergleichbare Technologien eingegangen und ihre Gemeinsamkeiten und Unterschiede zu diesem Projekt. Anschließend wird das Konzept und die Implementierung des Projekts beschrieben und das Ergebnis evaluiert. Der sechste Teil der Ausarbeitung widmet sich Möglichkeiten für zukünftige Projekte und Verbesserungen. Zuletzt wird ein Fazit gezogen und die Ergebnisse der Arbeit analysiert.

³<http://eexcess.eu/results/chrome-extension/>

⁴<http://www.europeana.eu/portal/>

2 Related Work

Rhodes [RM00] definiert JITIR-Agents als eine Klasse von Programmen, die dem Benutzer weiterführende Informationen basierend auf seinem lokalen Kontext anzeigen. Dabei beschränkt sich der überwachte Kontext meist auf die virtuelle Umgebung des Benutzers, wie E-Mail, Webseiten und geöffnete Dokumente. Als Kerneigenschaften von JITIR-Agents nennt Rhodes Selbstständigkeit, die Fähigkeit Informationen in einer leicht zugänglichen und gleichzeitig unaufdringlichen Weise darzustellen und das Bewusstsein über den Kontext des Benutzers [RM00]. Das im Zuge dieser Bachelorarbeit entworfene Programm (im Folgenden als Jarvis bezeichnet) erfordert vom Nutzer die explizite Aufforderung um nach Informationen zu suchen. Gründe für diese Entscheidung werden im dritten Kapitel näher betrachtet. Trotz dieses Widerspruchs zu Rhodes Definition lässt sich Jarvis am besten mit dieser Klasse von Programmen vergleichen. Warum diese Klassifizierung zutrifft wird im folgenden Abschnitt beschrieben.

2.1 Unterschiede und Gemeinsamkeiten zu JITIR-Agents

Studien haben gezeigt, dass schon eine kleine Steigerung des Aufwands, der betrieben werden muss um eine Aufgabe zu erfüllen, dazu führen kann, dass man die Aufgabe gar nicht erst ausführt [RM00]. Laut Miller reicht für die meisten Aufgaben eine Antwortverzögerung von mehr als zwei Sekunden aus, um die Nutzungshäufigkeit des dazugehörigen Programms zu vermindern [Mil68]. Längere Zeitintervalle erschweren es, den Kontext der gerade ausgeführten Aufgabe und die übergeordneten Aufgaben im Kurzzeitgedächtnis zu behalten. Nun wird der Fall betrachtet, dass das Lesen einer Webseite unterbrochen wird, um eine Suche mit einer Suchmaschine durchzuführen. Die eigentliche Aufgabe behält der Nutzer im Kurzzeitgedächtnis. Je länger die Suche dauert und je mehr er sich dazu von seiner eigentlichen Arbeit distanzieren muss, desto schwerer wird es wieder zur Hauptaufgabe zurück zu kehren. Wenn der Exkurs zur Suchmaschine schwerer wiegt als die Güte der erwarteten Resultate wird die Suche nicht durchgeführt [RM00].

Dieses Problem wird versucht mit JITIR-Agents zu beheben. Durch ihre proaktive Arbeitsweise muss der Nutzer seine Tätigkeit nicht mehr unterbrechen, sondern nur noch entscheiden ob er weiter Informationen sehen möchte oder nicht [RM00]. Beim Entwurf von Jarvis wurde entschieden, dass das Programm nicht völlig eigenständig nach Informationen sucht, sondern nur die Webseite in seine Paragraphen aufteilt. Der Nutzer kann dann entscheiden, ob er zu einem Paragraphen eine Suche durchführen möchte und diese dann per Klick starten. Wie bei einem JITIR-Agent wird die Suchanfrage automatisch aus den gewonnenen Kontextinformationen generiert. Da die Suche innerhalb weniger Millisekunden ausgelöst werden kann und sich der Nutzer dazu nicht von seiner eigent-

lichen Aufgabe distanzieren muss, bleibt die kognitive Belastung sehr gering. Allerdings entsteht auch so der Nachteil, den Jarvis mit JITIR-Agents gemein hat: Sie nutzen alle gefundenen Informationen für ihre Suchanfragen und können nicht zwischen relevanten und unwichtigen Suchwörtern unterscheiden [Rho00]. Automatisch gebaute Suchanfragen sind deshalb weniger exakt als Menschen-generierte [RM00]. Um dem entgegen zu wirken hat der Benutzer von Jarvis im Nachhinein noch die Möglichkeit, die Suche anzupassen und erneut abzuschicken.

Die zweite von Rhodes beschriebene Eigenschaft von JITIR-Agents, die Fähigkeit Informationen in einer leicht zugänglichen und gleichzeitig unaufdringlichen Weise darzustellen, hat auch beim Entwurf von Jarvis eine bedeutende Rolle gespielt. Für die Darstellung der Ergebnisse muss ein Mittelwert gefunden werden. Die zusätzlichen Elemente sollen den Benutzer nicht unnötig ablenken, allerdings will man die Seite um möglichst reichhaltige Informationen erweitern [Rho00]. Wie diese Problematik im Falle von Jarvis gelöst wurde wird im Implementierungs-Teil dieser Arbeit beschrieben.

Jarvis analysiert die geöffnete Webseite und teilt diese in Paragraphen ein. Wie JITIR-Agents ist er sich also über den lokalen Kontext des Benutzers bewusst und wertet diesen aus. Die dritte Voraussetzung für JITIR-Agents ist somit erfüllt.

Trotz der eingeschränkten Selbstständigkeit lässt sich Jarvis folglich am besten mit denen von Rhodes beschriebenen Programmen vergleichen. Auf diese Erkenntnis aufbauend wird nachfolgend Jarvis mit existierenden Implementierungen von JITIR-Agents verglichen.

2.2 Vergleich mit existierenden JITIR Agents

Es wurden einige JITIR-Agents in der Vergangenheit implementiert, die zwar aus technischer Hinsicht nicht mehr aktuell sind, die sich jedoch durch ihre genaue Untersuchung und Auswertung gut Vergleichen lassen.

2.2.1 Remembrance Agent

Der Remembrance Agent (RA) ist ein in den UNIX Texteditor EMACS-19 integriertes Programm, welches eine Liste von Dokumenten anzeigt, die für den Nutzer interessant sein könnten [RS96]. Die Informationen für die Vorschläge bezieht der RA aus der gerade geöffneten Datei. Wie Jarvis teilt er das Dokument in Bereiche ein, die er danach analysiert. Allerdings sind die Bereiche in diesem Fall keine Paragraphen sondern verschieden große "Räume" in denen er sucht. Von der Position des Cursors ausgehend untersucht er die nächsten zehn, die nächsten 50 und die nächsten 1000 Wörter des Dokuments und sucht zu jedem passende Vorschläge [RS96]. Die Vorschläge werden von einem Information Retrieval Programm im Hintergrund generiert, welches die Suchanfragen an

verschiedene, vom Nutzer konfigurierbare Datenbanken schickt. Genannte Datenbanken sind persönliche Email-Verzeichnisse, persönliche Notizen oder Datenbanken mit wissenschaftlichen Arbeiten.

Die gefundenen Ergebnisse werden dem Nutzer dann am unteren Fensterrand dargestellt. Mehrere Zeilen präsentieren die Vorschläge mit der höchsten Relevanz. Je nach Art der Quellen werden verschiedene Informationen angezeigt. So werden zum Beispiel bei wissenschaftlichen Arbeiten der Autor, das Datum und der Titel angezeigt und bei Zeitungsartikeln nur Herausgeber, Überschrift und Datum [RM00]. Dieses Design ermöglicht eine unaufdringliche Darstellung der Vorschläge, die leicht überblickt werden kann aber den Nutzer nicht von seiner Arbeit ablenkt [RS96]. Per Klick kann sich der Nutzer die Schlagwörter anzeigen lassen, die für das jeweilige Ergebnis verantwortlich sind [RM00]. Mit einer Tastenkombination und der Zeilennummer des Vorschlags kann er sich das dazugehörige, vollständige Dokument anzeigen lassen. Weiterhin hat der Nutzer die Möglichkeit, eine eigene Suchanfrage einzugeben um so explizit nach Inhalten zu suchen [RS96].

Mit Hilfe einer Nutzer-Evaluation konnte gezeigt werden, dass der RA nicht nur eine Alternative zu klassischen Suchmaschinen darstellt, sondern in vielen Bereichen sogar besser abscheidet. Durch die Nutzung fanden die Tester mehr relevante Dokumente zum geforderten Themengebiet und die anschließende Umfrage schnitt der RA besser ab als die der Kontrollgruppe zur Verfügung gestellte Suchmaschine [RM00].

2.2.2 Margin Notes

2.2.3 Watson

2.2.4 EEXCESS

2.3 Text Retrieval Algorithmen

Term Frequency/Inverse Document Frequency algorithm, Text rank

2.4 Unterschiede und Gemeinsamkeiten zu „Automatic help systems“ (z.B. Microsoft Office Assistant - \hat{z} Domain spezifisch)

Domain spezifisch vs. Domain unabhängig

3 Konzept

3.1 Warum kein Proactiver JITIR-Agent?

- ¿ API Limitierung und decrease cognitive load - ¿ Benutzer entscheidet ob er weitere Informationen erhalten möchte - ¿ Benutzer kann Suchanfrage erst anpassen (Nachteil von Margin Notes (Paper 4))

3.2 Anzeige der Ergebnisse

3.3 Erklärung der Such-Anfragen Generierung

3.4 Anpassen der Suchanfrage durch den Nutzer

3.5 Verbesserung der Suchanfrage z.B. durch maschinelles Lernen

4 Implementierung

4.1 Verwendung von AngularJS für alle Komponenten des Plugins

4.2 Bau der GUIs

- ¿ Darf den Benutzer nicht zu sehr ablenken - ¿ Ergebnisse müssen in der Nähe ihrer „Quelle“ angezeigt werden (proximity compatibility principle) - ¿ Benutzer muss klar zwischen Webseite und Augmentation unterscheiden können - ¿ bunt, auffälliges Design - ¿ Ramping interface: Mehr Benutzerinteraktionen führen zu mehr angezeigten Informationen (Erklärung der Stages)

4.3 Einbindung der REST-Services

5 Evaluierung?

Aufgaben die Benutzer mit EEXCESS Lösen mussten müssen sie jetzt mit Redesign lösen. Vergleich der Ergebnisse?

6 Future Work

6.1 Alternative Algorithmen zur Textanalyse

6.2 Implementierung einer automatischen

Suchanfragen-Verbesserung durch maschinelles Lernen

- Mehr kontextuelle Informationen miteinbeziehen
- Such-Profil des Nutzers erstellen

6.3 Verbesserung der Ergebniss-Güte

- durch Query Expansion
- durch Filtern der Ergebnisse (mehr Präzision da Ausbeute bei JITIR nicht so relevant)

6.4 Anpassung der Anwendung auf mobile Nutzung

7 Conclusion

- Steigerung der Effektivität und Produktivität von wissenschaftlichem Arbeiten
- Starke Effektivitätssteigerung durch Punkte aus Future Work möglich?

8 Quellenverzeichnis

- [AI] ALEXA INTERNET, Inc. 1996 2.: *Alexa*. <http://www.alexa.com/topsites>. – Zugriff: 13.05.2015, Archiviert mit WebCite®: <http://www.webcitation.org/5hgZUZacN>
- [Bou99] BOUVIN, Niels O.: Unifying strategies for Web augmentation. In: *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: returning to our diverse roots: returning to our diverse roots* ACM, 1999, S. 91–100
- [Día12] DÍAZ, Oscar: Understanding web augmentation. In: *Current Trends in Web Engineering*. Springer, 2012, S. 79–80
- [Law00] LAWRENCE, Steve: Context in web search. In: *IEEE Data Eng. Bull.* 23 (2000), Nr. 3, S. 25–32
- [Mil68] MILLER, Robert B.: Response time in man-computer conversational transactions. In: *Proceedings of the December 9-11, 1968, fall joint computer conference, part I* ACM, 1968, S. 267–277
- [Rho00] RHODES, Bradley J.: Margin notes: Building a contextually aware associative memory. In: *Proceedings of the 5th international conference on Intelligent user interfaces* ACM, 2000, S. 219–224
- [RM00] RHODES, Bradley J. ; MAES, Pattie: Just-in-time information retrieval agents. In: *IBM Systems journal* 39 (2000), Nr. 3.4, S. 685–704
- [RS96] RHODES, Bradley ; STARNER, Thad: Remembrance Agent: A continuously running automated information retrieval system. In: *The Proceedings of The First International Conference on The Practical Application Of Intelligent Agents and Multi Agent Technology*, 1996, S. 487–495

Anhang

GUI Screenshots

Unterkategorie, die nicht im Inhaltsverzeichnis auftaucht.

Erklärung

Hiermit versichere ich, dass ich meine Abschlussarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Datum:

.....

(Unterschrift)